(Edited Feb. 10, 2022)

Introduction

Assemble and simulate the pipelined datapath of your processor.

Datapath

The datapath of your microprocessor stores, accesses, and computes the operands of your instructions, and interfaces with the external modules. A good starting point for this exercise would be to look at the dataflow for each of your instructions and make sure that you have all the components necessary to do the desired computations. By now, you have designed most of the components needed to achieve the desired functionality. You may need to design tristate drivers and/or muxes and pipeline registers if you need them. You must also commit to either a tristate bus or multiplexor-based design; this decision together with the placement of modules on the datapath dictates the number of busses on your datapath. A natural next step is to assemble all of the modules together (in both the schematic and the layout) and simulate for functional and timing correctness. This requires that you present correct control signals to all of the control points in the simulation. These are input signals for now, since you do not have the control logic yet – however, if you'd like, you can get started on CAD8 by writing an instruction decoder for this CAD's testing purposes.

Procedure

1. **Include your previous CAD assignments**

After creating the CAD7 directory, open your cds.lib file and add the paths of your previous designs.
For example:
DEFINE RF /afs/umich.edu/class/eecs427/w22/group1/CAD3/RF
DEFINE ALU /afs/umich.edu/class/eecs427/w22/group1/CAD3/ALU

2. **Assemble & simulate the entire datapath in NCVerilog for functional verification**

Assemble the schematic of the datapath with the RF, ALU, Shifter, DMEM (RA1SH16x512), and other blocks (muxes, etc), and use NCVerilog to verify the functionality. You will have a lot of control signals to force when doing NCVerilog simulation, and it may be simpler if you have a simple controller/decoder in Verilog to provide those input stimuli (you will also get a head start on CAD8 this way). Run at least one simulation each for all instructions (total 19, no Bcond, Jcond, or JAL). Pay careful attention to the write-back portion of your execute pipeline stage. List and describe all of the control signals for the datapath in your README. Your

description should include functional specifics for your individual components. You may include a truth-table if you'd like. This will be needed for CAD8 and will help the grader understand your verification.

3. **Assemble datapath layout**

Connect each module manually and add power rings around your datapath.

4. **Update the delay for each module & add the total delay to the datapath output stage**

The datapath is rather large and running simulations on the whole design is unfeasible. Instead, with the datapath complete, you may extract the load capacitance on each of your module outputs by looking at the PEX result (caliber view). Place the load capacitance values into earlier circuit simulations of each datapath module for the outcap and re-simulate your modules. This should give you more accurate delays for that particular module. After updating the delay information of each module, add the worst-case total delay to the output stage of your datapath and simulate datapath again in NCVerilog.

Submission

Please refer to the submission guidelines mentioned in the CAD1 assignment, as they apply to all CAD assignments. Do not lose points over trivial mistakes like incorrect file names, etc!

For CAD7, you need to submit the following:

- Datapath:
  - Schematic of the entire datapath
  - Layout of the datapath. You should extend all pins to the boundary and wire VDD and VSS pins together on the rings.
  - Clean DRC/LVS report.
  - NCVerilog simulation waveform of the entire datapath. At least one simulation each for all instructions.
  - NCVerilog simulation waveform with worst case delay added. Simulate one instruction that triggers your worst-case delay.
- README:
  - List of all the control signals you need with descriptions. You may include a truth-table for them.
  - Path to the testfixture.verilog
  - All input pin cap
  - Create a table in your README comparing the old/new delay numbers for each block (RF, ALU, Shifter, etc) and the total delay by adding them up

- Documentation is important, so make sure you comment your report well. Important considerations that went into the design, and extra features, if any, should be documented.

Deadline

You must complete CAD7 by **Mar. 10, 2022** by **11pm**.
Do not modify any files in your CAD directory after that time.